

qRAM and the Quantum Data-Loading Bottleneck

An Open Problem at the Theory–Engineering Boundary

Paolo Bernardi

Department of Computer Science
University of Pisa

“Mauriana Pesaresi” Seminar Series 2026

Outline

- 1 Motivation: The Data-Loading Bottleneck
- 2 The qRAM Abstraction
- 3 Bucket-Brigade Architecture
- 4 Circuit-Level Analysis
- 5 Noise, Fault Tolerance, and the Critique
- 6 Alternative Approaches and Workarounds
- 7 Open Problems and Conclusions

Quantum Computing in a Nutshell

Qubit: the quantum analogue of a classical bit. While a bit is 0 or 1, a qubit can be in a *superposition*:

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle, \quad |\alpha|^2 + |\beta|^2 = 1$$

Key concepts for this talk:

- ▶ **Superposition:** a system of n qubits can represent 2^n states simultaneously:
 $|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle$
- ▶ **Entanglement:** qubits can be correlated in ways with no classical counterpart; measuring one affects the others
- ▶ **Unitary gates:** quantum operations are reversible linear transformations (matrices U with $U^\dagger U = I$)
- ▶ **Measurement:** collapses superposition; outcome $|0\rangle$ with probability $|\alpha|^2$ and outcome $|1\rangle$ with probability $|\beta|^2$; information is *irreversibly extracted*
- ▶ **Decoherence:** interaction with the environment destroys superposition — the central engineering challenge

Classical vs. Quantum Memory Access

Classical RAM

- ▶ Address $\rightarrow$ Data (deterministic)
- ▶ $\mathcal{O}(1)$ per random access
- ▶ Mature, scalable technology

Quantum algorithms need

- ▶ Superposition over addresses:

$$\sum_i \alpha_i |i\rangle |0\rangle \longrightarrow \sum_i \alpha_i |i\rangle |x_i\rangle$$

- ▶ Coherent, reversible access
- ▶ $\mathcal{O}(1)$ query complexity

The bottleneck

Many celebrated quantum speedups (Grover search, HHL, quantum machine learning) *assume* this $\mathcal{O}(1)$ coherent access as a free oracle. If the oracle is expensive, the speedup may vanish.

Where qRAM Appears in the Literature

- ▶ **Grover's search** [5]: oracle access to a function $f: \{0, 1\}^n \rightarrow \{0, 1\}$
- ▶ **HHL algorithm** [6]: coherent access to matrix entries for linear systems
- ▶ **Quantum recommendation** [7]: $\mathcal{O}(\text{polylog}(mn))$ for preference matrices
- ▶ **Quantum machine learning** broadly [8]: amplitude encoding of classical data

Key question

What is the *actual cost* of implementing the oracle? If $\mathcal{O}(N)$ gates are needed, the quantum advantage can reduce to constant-factor.

Formal Definition

A qRAM for a dataset $\{x_0, x_1, \dots, x_{N-1}\}$, $x_i \in \{0, 1\}^m$, implements the unitary:

$$U_{\text{qRAM}} |i\rangle |0\rangle^{\otimes m} = |i\rangle |x_i\rangle \quad \forall i \in \{0, \dots, N-1\}$$

Parameters of interest:

- ▶ **Address register:** $n = \lceil \log_2 N \rceil$ qubits
- ▶ **Query complexity:** number of calls to U_{qRAM}
- ▶ **Gate complexity:** gates needed to *realize* U_{qRAM}
- ▶ **Depth:** circuit depth (time cost)
- ▶ **Width:** total qubits (space cost)

Oracle vs. Circuit vs. HW Perspective

Oracle model

$\mathcal{O}(1)$ query
“black box”

Circuit model

$\mathcal{O}(\cdot)$ gates/depth
explicit decomposition

Hardware

physical qubits,
error rates, connectivity

- ▶ Algorithmic papers typically work at the **oracle** level.
- ▶ Complexity-theoretic cost hides in the **circuit** compilation.
- ▶ Real feasibility depends on the **hardware** level.

The gap between these three levels is the core of the qRAM debate.

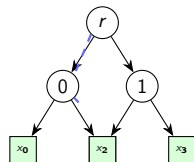
The GLM Bucket-Brigade Proposal

Giovannetti, Lloyd, and Maccone [1, 2]:

Key idea: binary routing tree

- ▶ N memory cells at the leaves
- ▶ $N - 1$ internal *routing nodes*
- ▶ Each node: tri-state qubit $\{|w\rangle, |0\rangle, |1\rangle\}$
(wait, route-left, route-right)
- ▶ Address qubits activate one root-to-leaf path

Claimed advantage: only $\mathcal{O}(\log N)$ nodes are *active* per query $\Rightarrow$ noise affects $\mathcal{O}(\log N)$ components



$n = 2$ address bits, $N = 4$ cells.

Blue path: address $|01\rangle$.

Bucket-Brigade: How It Works

Query protocol (for address $|a_{n-1} \cdots a_1 a_0\rangle$):

- 1 All routing nodes start in $|w\rangle$ (wait).
- 2 **Route-in phase**: for $k = n - 1, \dots, 0$, the address qubit a_k sets the routing node at level k on the active path to $|a_k\rangle$.
- 3 At the leaf: a SWAP transfers the memory content to the bus qubit.
- 4 **Route-out phase**: reverse the routing (uncompute the path).

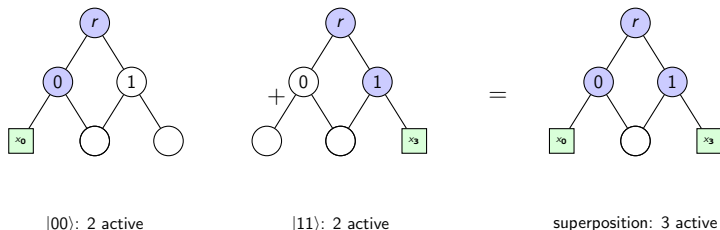
In superposition: $\sum_i \alpha_i |i\rangle |0\rangle \xrightarrow{U_{\text{BB}}} \sum_i \alpha_i |i\rangle |x_i\rangle$

Resources (per query):

- ▶ **Width**: $\mathcal{O}(N)$ routing qubits + n address qubits + bus
- ▶ **Depth**: $\mathcal{O}(n) = \mathcal{O}(\log N)$
- ▶ **Active components**: $\mathcal{O}(n)$ (the single root-to-leaf path)

Superposition of Paths

Crucial subtlety: in superposition, *all* $N - 1$ routing nodes participate in the global entangled state, even though each branch of the superposition activates only $\mathcal{O}(\log N)$.



This distinction between *per-branch* active count and *global* entangled state is central to the noise analysis debate.

Compiling qRAM to Gates

Naïve approach: decompose U_{qRAM} as a sequence of controlled operations (multi-controlled CNOTs).

For N data items of m bits:

- ▶ **Gate count:** $\mathcal{O}(Nm)$ in general [9]
- ▶ **Depth:** $\mathcal{O}(n)$ with parallelism (fan-out) [10]
- ▶ **T-gate cost** (fault-tolerant): dominant resource; each Toffoli $\rightarrow$ multiple T-gates

The cost is $\mathcal{O}(N)$, not $\mathcal{O}(\log N)$

Even with bucket-brigade structure, the *total* gate count for loading N arbitrary data items is $\Omega(N)$. The $\mathcal{O}(\log N)$ claim applies to **depth** (with $\mathcal{O}(N)$ width), not total work.

QROM: A Read-Only Alternative

Quantum Read-Only Memory (QROM) [11]:

- ▶ Hardcodes classical data as a *circuit* (no persistent quantum memory)
- ▶ Implements the same unitary $|i\rangle |0\rangle \rightarrow |i\rangle |x_i\rangle$
- ▶ Uses unary iteration or select-swap networks

Trade-offs:

	qRAM (BB)	QROM (circuit)
Depth	$\mathcal{O}(\log N)$	$\mathcal{O}(N)$ or $\mathcal{O}(\sqrt{N})$ ¹
Width	$\mathcal{O}(N)$ routing qubits	$\mathcal{O}(n + m)$
Persistent state	Yes (routing nodes)	No
Error model	Routing noise	Standard gate errors

¹With space-time trade-off [10].

Architecture-Aware State Preparation

Berti & Ghisoni [16]: explicit polylogarithmic state preparation on *hardware* BBQRAM—a constructive counterpoint to the “hidden $\mathcal{O}(N)$ cost” concern.

Construction:

- ▶ Classical preprocessing: build a *segment tree* of squared norms of matrix entries
- ▶ **Memory layout** embedding the tree into BBQRAM cells (sibling pairs packed per cell, contiguous addressing per level)
- ▶ **Quantum retrieval primitives** that access each tree level in superposition

Result: encode $A \in \mathbb{R}^{M \times N}$ in

- ▶ **Time:** $\mathcal{O}(\log_2^2(MN))$
- ▶ **Output register:** $\Theta(\log_2(MN))$ qubits; constant working qubits (fixed precision)
- ▶ **Memory cells:** $\mathcal{O}(MN)$ within BBQRAM

Significance

Sharpens the implicit $\text{polylog}(MN)$ bound of Kerenidis–Prakash [7] into an explicit $\log^2$ complexity, *aware* of the underlying QRAM architecture. Assumes a functioning HW BBQRAM.

GLM's Noise-Resilience Argument

Original claim [2]:

"Because only $\mathcal{O}(\log N)$ routing nodes are active per query, decoherence scales as $\mathcal{O}(\log N)$, not $\mathcal{O}(N)$."

The argument:

- ▶ Inactive (wait-state) nodes do not interact with the bus
- ▶ Phase errors on $|w\rangle$ nodes are benign
- ▶ Fidelity: $F \sim 1 - \mathcal{O}(\varepsilon \log N)$ where ε is per-node error rate

This is the key claim that, if true, would make large-scale qRAM potentially feasible even *without* full quantum error correction on the routing nodes.

Hann et al.: The Counter-Argument

Hann, Kang, et al. [3]:

Main result

Under a generic (non-fine-tuned) noise model, the query fidelity of bucket-brigade qRAM scales as:

$$F \sim (1 - \varepsilon)^{\mathcal{O}(N)}$$

i.e., **exponential** decay in N , not polylogarithmic.

Why?

- ▶ Even “inactive” nodes are *entangled* with the routed state
- ▶ A bit-flip on a wait-state node can *misroute* future or concurrent superposition branches
- ▶ The noise resilience of GLM relies on a restrictive error model (only dephasing, no bit-flip on routing qubits)

The Fidelity Debate: Summary

Noise model	Fidelity scaling	Source
Pure dephasing only	$1 - \mathcal{O}(\varepsilon \log N)$	GLM [2]
Depolarizing / generic	$(1 - \varepsilon)^{\mathcal{O}(N)}$	Hann+ [3]
With QEC on routers	depends on code distance	open

Implications:

- ▶ If generic noise applies $\Rightarrow$ qRAM needs **full fault-tolerant error correction** on all $\mathcal{O}(N)$ nodes
- ▶ This erases the practical advantage of the BB architecture
- ▶ The resource cost becomes comparable to (or worse than) QROM

Estimates for loading $N = 10^6$ data items [10, 4]:

- ▶ **Physical qubits** (fault-tolerant BB): 10^7 – 10^9 depending on code distance
- ▶ **Latency**: $\sim \mu s$ per query (optimistic) vs. ms-scale quantum algorithms
- ▶ **T-factory overhead**: thousands of ancilla factories running in parallel

Perspective

Current quantum computers: $\sim 10^3$ physical qubits.

The gap is 4–6 orders of magnitude for a practically useful qRAM.

Several research directions avoid the qRAM assumption:

- ① **Explicit state preparation:** encode data into amplitudes via circuits; $\mathcal{O}(N)$ depth but no persistent quantum memory [12]
- ② **Block encoding / LCU:** access matrix entries via structured decompositions; powerful for sparse or structured data [13]
- ③ **Quantum-inspired classical algorithms:** Tang's dequantization program; classical algorithms matching quantum speedups when ℓ_2 -sampling is available [14]
- ④ **Variational / hybrid methods:** data stays classical; quantum circuit acts on parametrized ansatz only

The Dequantization Wake-Up Call

Tang (2019) [14]: classical algorithm for recommendation matching the quantum algorithm of Kerenidis and Prakash [7].

Key insight: the quantum speedup in many QML algorithms comes from *the qRAM assumption* (efficient ℓ_2 -sampling), not from quantum interference per se.

If a classical data structure supports ℓ_2 -sampling in $\mathcal{O}(\text{polylog}(N))$, the quantum advantage vanishes.

Lesson

qRAM is not “free infrastructure”, it is an **algorithmic assumption** that can shift the complexity boundary between classical and quantum.

Recent Progress: Architecture-Aware State Preparation

Berti and Ghisoni (2025) [16]: a concrete efficient state-preparation algorithm for a real matrix $A \in \mathbb{R}^{M \times N}$ under the **hardware BBQRAM** model.

Key idea: embed a classical *Segment Tree* of squared norms into BBQRAM memory cells, with a layout that preserves the tree's hierarchy and supports logarithmic-time retrieval in superposition.

Result

Amplitude encoding of A in

time $\mathcal{O}(\log_2^2(MN))$, $\Theta(\log_2(MN))$ qubits, $\mathcal{O}(MN)$ memory cells

with a *constant* number of working qubits (fixed precision) and $\mathcal{O}(MN)$ classical preprocessing.

Sharpens the earlier $\mathcal{O}(\text{polylog}(MN))$ bound of Kerenidis and Prakash [7] by making the BBQRAM routing overhead explicit, and illustrates the case for **architecture-aware** algorithm design.

Open Problems

- ➊ **Noise models:** what is the *realistic* error model for idle routing qubits in a bucket-brigade? Experimental characterization needed.
- ➋ **Hybrid architectures:** partial error correction on routing nodes? Selective protection of critical tree levels?
- ➌ **Lower bounds:** is $\Omega(N)$ total work (gates $\times$ depth) provably required for arbitrary data loading?
- ➍ **Structured data:** for specific data distributions (sparse, low-rank, efficiently computable), can we beat the generic $\mathcal{O}(N)$ cost?
- ➎ **Application-specific trade-offs:** given a concrete algorithm (e.g., HHL for a class of matrices), when does the qRAM overhead erase the quantum speedup?

Conclusions

- ▶ qRAM is **not a solved problem**: it sits at the boundary of theory, architecture, and engineering.
- ▶ The bucket-brigade architecture is **elegant but fragile**: noise resilience depends critically on the error model.
- ▶ **Circuit-level cost** is $\Omega(N)$ in general, eroding the promised exponential quantum speedups.
- ▶ The qRAM debate forces us to be **honest about assumptions**: oracle-level complexity results need circuit-level validation.
- ▶ **Active research area**: alternatives (QROM, block encoding, dequantization) are reshaping which quantum speedups survive scrutiny.

Thank you. . . Questions?

References I

-  V. Giovannetti, S. Lloyd, and L. Maccone, “Quantum random access memory”, 2008.
-  V. Giovannetti, S. Lloyd, and L. Maccone, “Architectures for a quantum random access memory”, 2008.
-  C. T. Hann, G. Lee, S. M. Girvin, and L. Jiang, “Resilience of quantum random access memory to generic noise”, 2021.
-  S. Jaques and A. Rattew, “QRAM: A survey and critique”, 2025.
-  L. K. Grover, “A fast quantum mechanical algorithm for database search”, 1996
-  A. W. Harrow, A. Hassidim, and S. Lloyd, “Quantum algorithm for linear systems of equations”, 2009
-  I. Kerenidis and A. Prakash, “Quantum recommendation systems,”, 2016
-  J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, “Quantum machine learning”, 2017.

References II

-  D. K. Park, F. Petruccione, and J.-K. K. Rhee, “Circuit-based quantum random access memory for classical data”, 2019.
-  C. T. Hann, G. Lee, S. M. Girvin, and L. Jiang, “Hardware-efficient quantum random access memory with hybrid quantum acoustic systems”, 2019.
-  R. Babbush, C. Gidney, D. W. Berry, N. Wiebe, J. McClean, A. Paler, A. Fowler, and H. Neven, “Encoding electronic spectra in quantum circuits with linear T complexity,”, 2018.
-  V. V. Shende, S. S. Bullock, and I. L. Markov, “Synthesis of quantum logic circuits”, 2005.
-  A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, “Quantum singular value transformation and beyond”, 2019
-  E. Tang, “A quantum-inspired classical algorithm for recommendation systems”, 2019
-  S. Arunachalam and R. de Wolf, “Optimal quantum sample complexity of learning algorithms”, 2018.
-  A. Berti and F. Ghisoni, “Efficient quantum state preparation with bucket brigade QRAM”, 2025